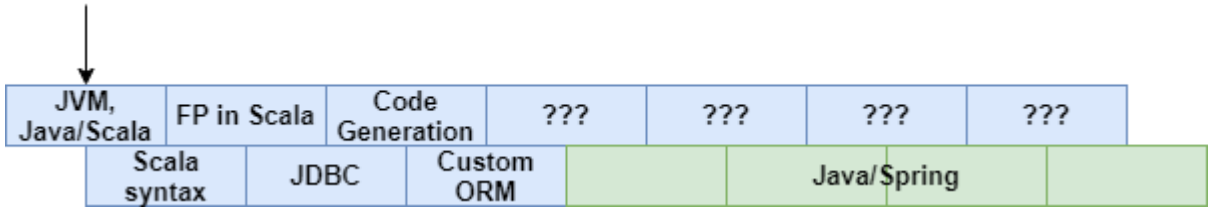
JVM, Java/Scala basics

ПЕРВАЯ ЛЕКЦИЯ

План курса

- Scala syntax
- Java syntax
- FP in scala
- JDBC (scalikeJDBC/Hybernate)
- HTTP server (Akka/Spring)
- Code generation/parsing (scalameta)
- Scala 3 - ?
- Scala plugin - ?
- Elastic search
- JMS/XML
- Java mail 



```java
package ru.itmo.backend_2021;

public class JavaMain {
  public static void main(String[] args) {
    System.out.println("Hello Java");
  }
}
```

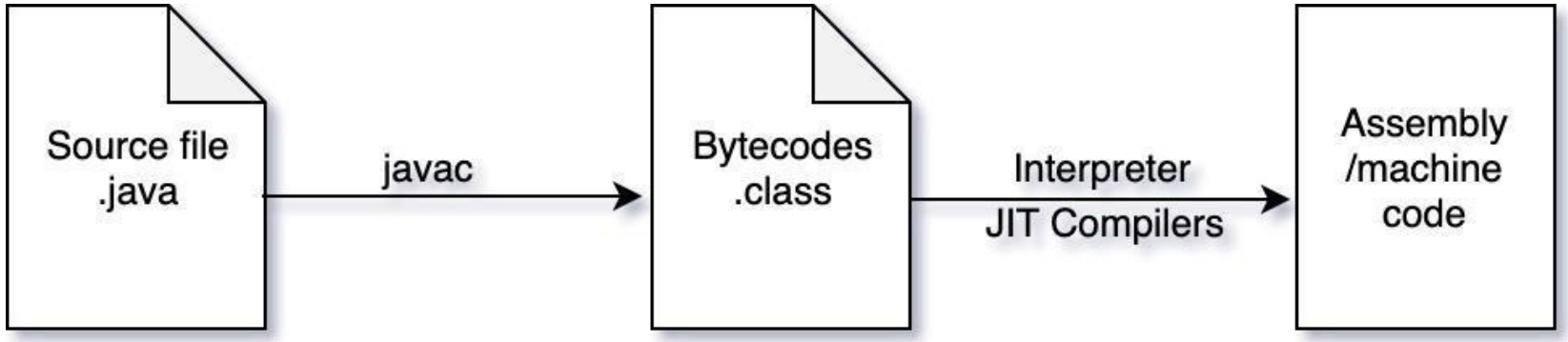
Java program

```java
public class WordCountJava {
  public static void main(String[] args) {
    StringTokenizer st = new StringTokenizer(args[0]);
    Map<String, Integer> map = new HashMap<>();
    while (st.hasMoreTokens()) {
      String word = st.nextToken();
      Integer count = map.get(word);
      if (count == null)
        map.put(word, 1);
      else
        map.put(word, count + 1);
    }
    System.out.println(map);
  }
}
```

Java program 2

Java compiler - JavaC

- Write once, run everywhere
- Converts Java code to Bytecode without any optimizations*
- Bytecode is intermediate language



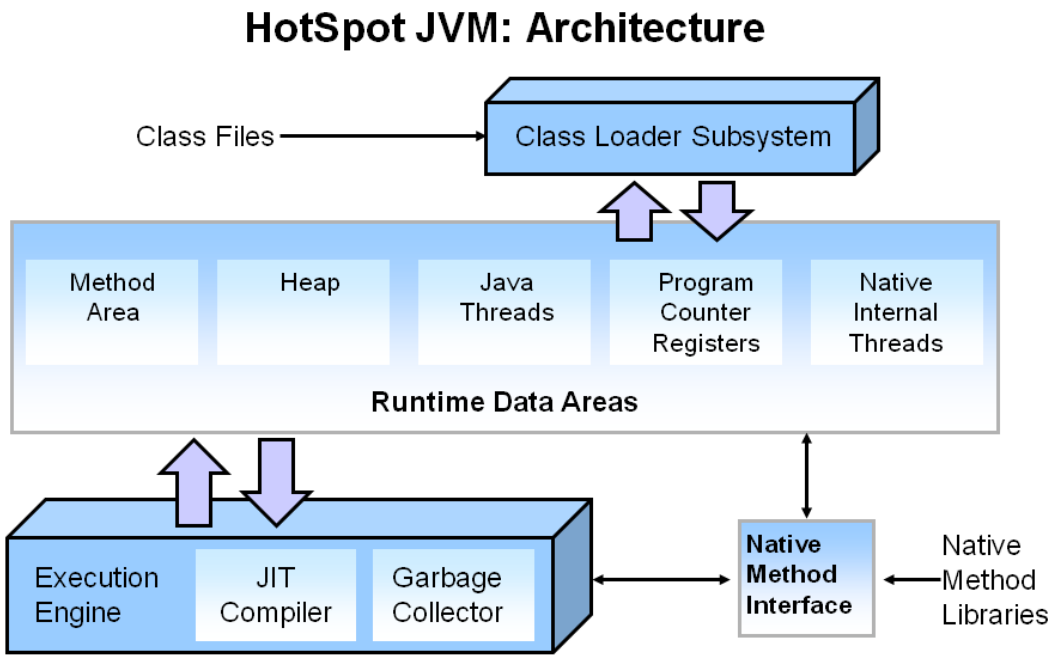
```java
public class WordCountJava {
  public static void main(String[] args) {
    StringTokenizer st = new StringTokenizer(args[0]);
    Map<String, Integer> map = new HashMap<>();
    while (st.hasMoreTokens()) {
      String word = st.nextToken();
      Integer count = map.get(word);
      if (count == null)
        map.put(word, 1);
      else
        map.put(word, count + 1);
    }
    System.out.println(map);
  }
}
```

```java
public class WordCountJava {
  public WordCountJava() {
  }

  public static void main(String[] args) {
    StringTokenizer st = new StringTokenizer(args[0]);
    HashMap map = new HashMap();

    while(st.hasMoreTokens()) {
      String word = st.nextToken();
      Integer count = (Integer)map.get(word);
      if (count == null) {
        map.put(word, 1);
      } else {
        map.put(word, count + 1);
      }
    }

    System.out.println(map);
  }
}
```

Java compiler - example

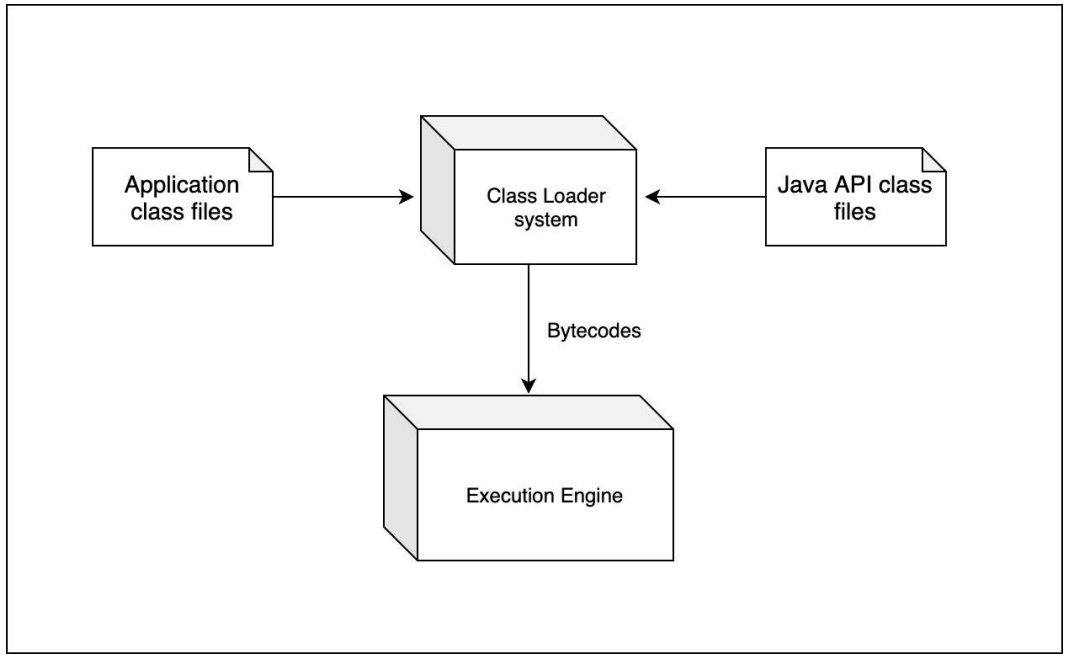
Java Virtual Machine

- Class loader
- Runtime Data Areas
- Execution Engine
- Native methods



JVM – class loader

- Loading
- Linking
  - Verification
  - Preparation
  - Resolution
- Initialization



```java
package ru.itmo.backend_2021;

public class JavaMain {
  public static void main(String[] args) {
    System.out.println("Hello Java");
  }
}
```

```scala
package ru.itmo.backend_2021

object ScalaMain {
  def main(args: Array[String]): Unit = {
    println("Hello Scala")
  }
}
```

Scala program

```java
public class WordCountJava {
  public static void main(String[] args) {
    StringTokenizer st = new StringTokenizer(args[0]);
    Map<String, Integer> map = new HashMap<>();
    while (st.hasMoreTokens()) {
      String word = st.nextToken();
      Integer count = map.get(word);
      if (count == null)
        map.put(word, 1);
      else
        map.put(word, count + 1);
    }
    System.out.println(map);
  }
}
```

```scala
object WordCountScala extends App {
  println(
    args(0)
      .split( regex = " ")
      .groupBy(identity)
      .transform((_, v) => v.length)
  )
}
```

Scala program

Scala compiler - ScalaC

- Scala is syntactic sugar for Java
- ScalaC compiles scala code into java code

```scala
object WordCountScala extends App {
  println(
    args(0)
      .split( regex = " ")
      .groupBy(identity)
      .transform((_, v) => v.length)
  )
}
```

Scala compiler example

SBT, Apache Maven

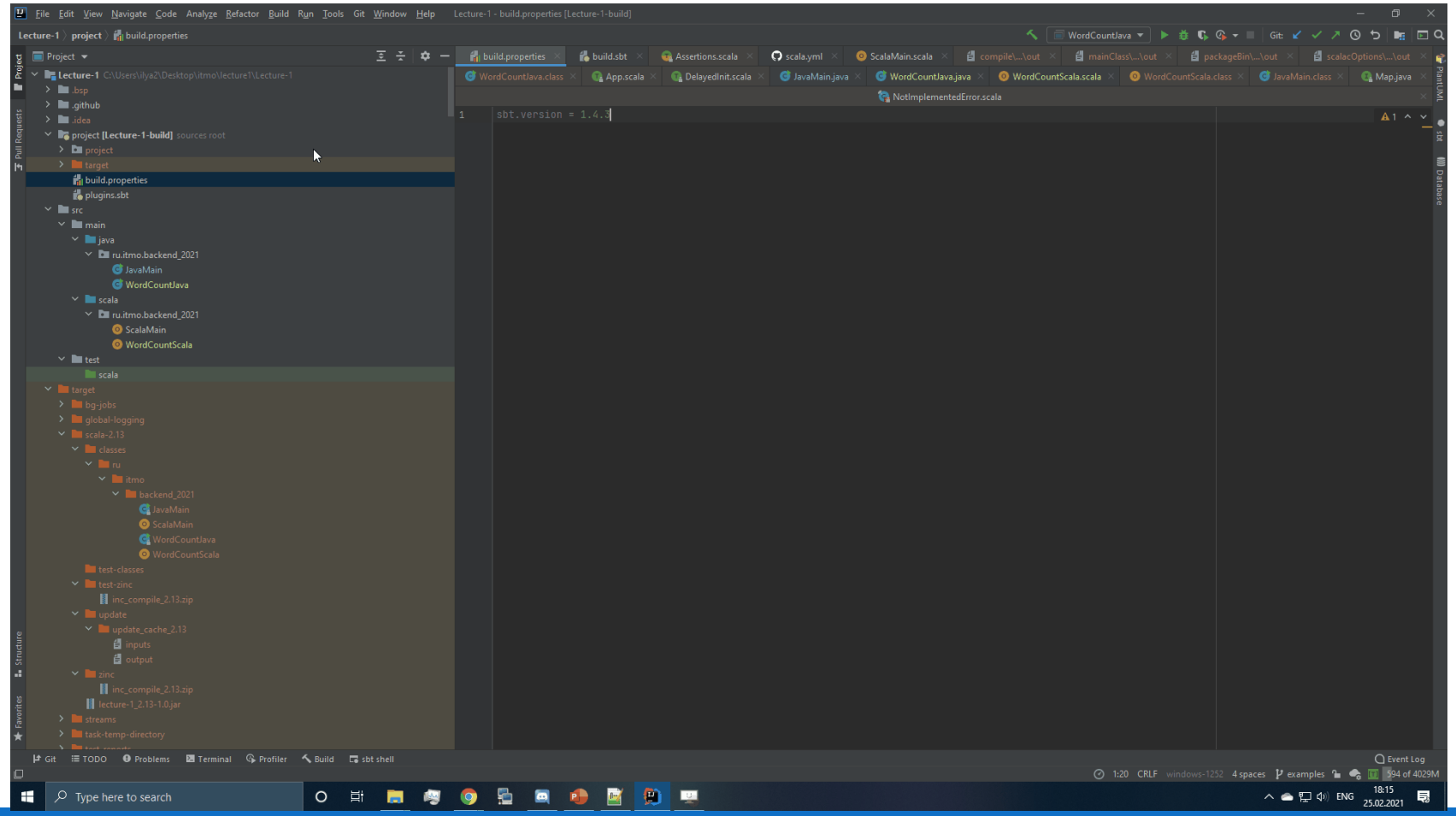
Build tool

Continuous compilation

Incremental testing and compilation

Dependency management

Setting up basic project



Useful links

JVM class loader - https://habr.com/ru/company/otus/blog/468193/

JVM class structure - https://habr.com/ru/company/otus/blog/478584/

Scala compiler steps - https://www.iteratorshq.com/blog/scala-compiler-phases-with-pictures/

Lecture 1 homework - https://github.com/Backend-ITMO-2021/Lecture-1